Authors: Andrei Tezhyk, Yaraslau Shemet

Student indexes: 101032, 108887

NLP - Dictionary Based Sentiment Analysis or Sentiment Analysis with ML

Warsaw 2024

NLP – Sentiment analysis with ML

1. Definitions of NLP and Sentiment analysis

Natural Language Processing, or NLP, is a field of artificial intelligence (AI) focused on enabling computers to understand, interpret, and generate human language in a way that is both meaningful and useful. It combines computational linguistics (rule-based language modeling) with statistical, machine learning, and deep learning models to process language data.

NLP is used in many applications that impact daily life and work. It powers virtual assistants like Siri, Alexa, and Google Assistant, which interpret spoken commands to carry out tasks. It is also essential in machine translation tools, such as Google Translate, which convert text from one language to another, and in text analysis tools that filter and classify content. Customer service chatbots rely on NLP to engage users and address common questions. In fields like healthcare, NLP processes medical records to identify key information, while in finance, it analyzes market sentiment by examining news and social media. NLP also supports sentiment analysis, summarization, and even creative uses like generating articles or stories. With its capacity to make human language accessible to machines, NLP continues to transform numerous industries, automating tasks, and enhancing interactions between people and technology.

Sentiment analysis is a technique used to automatically determine the emotional tone or opinion expressed in a piece of text. Often categorized as positive, negative, or neutral, sentiment analysis can be applied to a range of data sources, such as social media posts, product reviews, customer feedback, and news articles. The goal is to understand the sentiment behind the words, providing valuable insights into public opinion, customer satisfaction, or the emotional impact of content.

Businesses and researchers use sentiment analysis to gauge trends, monitor brand reputation, track customer feedback, and even predict stock market movements based on public sentiment. With machine learning continually advancing, sentiment analysis is becoming increasingly accurate and context-aware, making it a powerful tool.

2. Dictionary Based Sentiment Analysis vs Sentiment Analysis with ML

There are two frequently used approaches to NLP: Dictionary based sentiment analysis and Sentiment analysis with ML.

First one, Dictionary based sentiment analysis relies on a predefined lexicon or dictionary, where words are labeled with sentiment values (such as "positive," "negative," or "neutral"). The process involves scanning text for these words and calculating sentiment based on their occurrence and scores.

Its advantages are:

- It's easy to interpret the results since each score is derived directly from dictionary entries.
- Processing is faster since it doesn’t require training a model.

- It doesn’t require large training data, making it suitable even for small datasets or specific domains.

However it also has some limitations:

- It’s limited to words in the dictionary, making it difficult to adapt to new words, phrases, or contextual changes in sentiment.
- It struggles with context, such as sarcasm, slang, or domain-specific language, since dictionary-based approaches are not context-aware.

The second approach, Sentiment analysis with ML, involves training a model on labeled data (text samples marked with sentiment labels like "positive" or "negative") to learn sentiment patterns. It uses algorithms to predict sentiment based on features learned from the data.

Such model:

- can adapt to new language trends, context, and domain-specific language.
- It is context sensitive.
- Is generally more accurate.

However, it is also:

- computationally expensive,
- requires large datasets,
- can be hard to interpret.

In our project we went with both approaches - dictionary-based and machine learning. We obtained scores based on a predefined dictionary first and then implemented two different ML models and compared them.

3. How does NLP with ML work?

The process of sentiment analysis involves several steps. It begins with preprocessing the text data, where the system cleans the input by removing unnecessary characters, punctuation, and, often, stop words (common but unimportant words like "the" and "is"). After preprocessing, the text is then tokenized, breaking it into smaller pieces or words that a machine learning model can process. For machine learning-based sentiment analysis, these tokens are usually transformed into numerical values through vectorization, allowing the model to analyze and classify them.

Once the data is prepared, machine learning algorithms, typically trained on large datasets labeled with sentiment information, are applied. Commonly used algorithms for sentiment analysis include Naïve Bayes, support vector machines, and more recently, neural networks and transformers. For each piece of input, the model evaluates the probability of each sentiment category based on learned patterns from the training data.

After processing, the model outputs a sentiment score or classification, indicating whether the text is positive, negative, or neutral. Some models go further by providing intensity scores, which indicate how strongly positive or negative the sentiment is.

4. How does a Dictionary based approach work?

Dictionary-based sentiment analysis works by using a predefined lexicon or dictionary of words that are associated with specific sentiments, such as positive, negative, or neutral. The core idea is to match words in a given text to those in the dictionary and infer the overall sentiment based on the identified matches.

This method begins with constructing or utilizing an existing sentiment lexicon, which lists words along with their sentiment scores or labels. These scores can indicate polarity (positive or negative) and sometimes intensity. For example, words like "happy" or "excellent" may have a high positive score, while "sad" or "terrible" may be marked strongly negative. Neutral words or those with ambiguous sentiment might not be included or are assigned a score near zero.

When a piece of text is analyzed, the algorithm tokenizes it into individual words or phrases. Each token is compared against the sentiment dictionary to check for matches. If a word in the text appears in the dictionary, its associated sentiment score is retrieved. The system then aggregates these scores across the entire text, often by summing them or calculating an average, to produce an overall sentiment score or classification.

5. Dataset

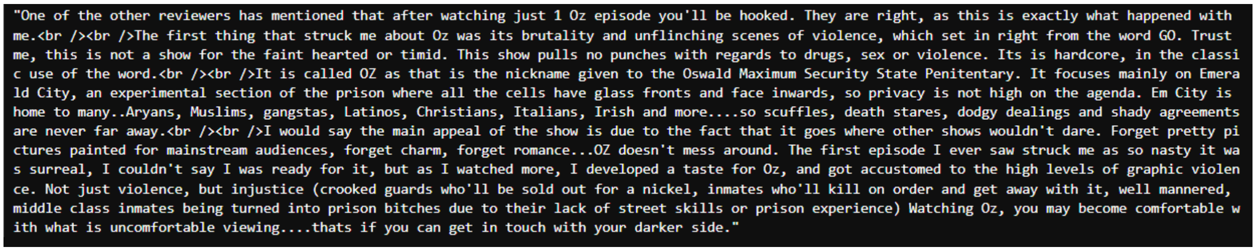
For this project we decided to use well-known “IMDB Dataset of 50K Movie Reviews” dataset found on Kaggle:
https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews/data (access: 11.10.2024)
It contains 50 000 observations in two columns:

1) review,
2) sentiment (with values positive / negative).

Data is balanced and does not contain missing values. There are some duplicated reviews in the dataset, they are either single-word reviews or duplicates for balancing the data.

6. Text preprocessing

First step in our analysis is data investigation. Some reviews contain HTML tags due to the scraped nature of the data from the IMDB webpage.



```
"One of the other reviewers has mentioned that after watching just 1 Oz episode you'll be hooked. They are right, as this is exactly what happened with
me.<br /><br />The first thing that struck me about Oz was its brutality and unflinching scenes of violence, which set in right from the word GO. Trust
me, this is not a show for the faint hearted or timid. This show pulls no punches with regards to drugs, sex or violence. Its is hardcore, in the classi
c use of the word.<br /><br />It is called OZ as that is the nickname given to the Oswald Maximum Security State Penitentary. It focuses mainly on Emera
ld City, an experimental section of the prison where all the cells have glass fronts and face inwards, so privacy is not high on the agenda. Em City is
home to many..Aryans, Muslims, gangstas, Latinos, Christians, Italians, Irish and more....so scuffles, death stares, dodgy dealings and shady agreements
are never far away.<br /><br />I would say the main appeal of the show is due to the fact that it goes where other shows wouldn't dare. Forget pretty pi
ctures painted for mainstream audiences, forget charm, forget romance...OZ doesn't mess around. The first episode I ever saw struck me as so nasty it wa
s surreal, I couldn't say I was ready for it, but as I watched more, I developed a taste for Oz, and got accustomed to the high levels of graphic violen
ce. Not just violence, but injustice (crooked guards who'll be sold out for a nickel, inmates who'll kill on order and get away with it, well mannered,
middle class inmates being turned into prison bitches due to their lack of street skills or prison experience) Watching Oz, you may become comfortable w
ith what is uncomfortable viewing....thats if you can get in touch with your darker side."
```

HTML elements are handled by **BeautifulSoap** library. In this way plain text is obtained from a review. We decided to use a ready-to-use solution due to the fact that usage of regular expressions can be error-prone in a task where investigation of every possible scenario would be time-consuming.

In the next step special characters (like #) and digits are removed from a review. Using the regular expression pattern **r"[^a-zA-Z\s']"** only spaces and letters are left in review. The only non-whitespace symbol allowed is apostrophe to be handled by the stopword dictionary.

As a part of Natural Language Processing tokenization of the reviews should be applied. Tokenization is the process of breaking down sentences into smaller, more manageable units. The purpose of this procedure is to represent text in a manner that is meaningful for machines without losing its context. Pattern recognition becomes easier with the simpler text. The task is to split the review into separate words using **word_tokenize** function from the NLTK library. In our case ‘word’ level of granularity is sufficient for sentiment analysis.

For NLP modelling and especially sentiment analysis, **stopwords** do not add much value. By removing them we focus only on the most important information avoiding for instance ‘i’, ‘and’, ‘our’. The case of the words is ignored to ensure that all of the stopwords are correctly removed. The NLTK stopwords dictionary is used.

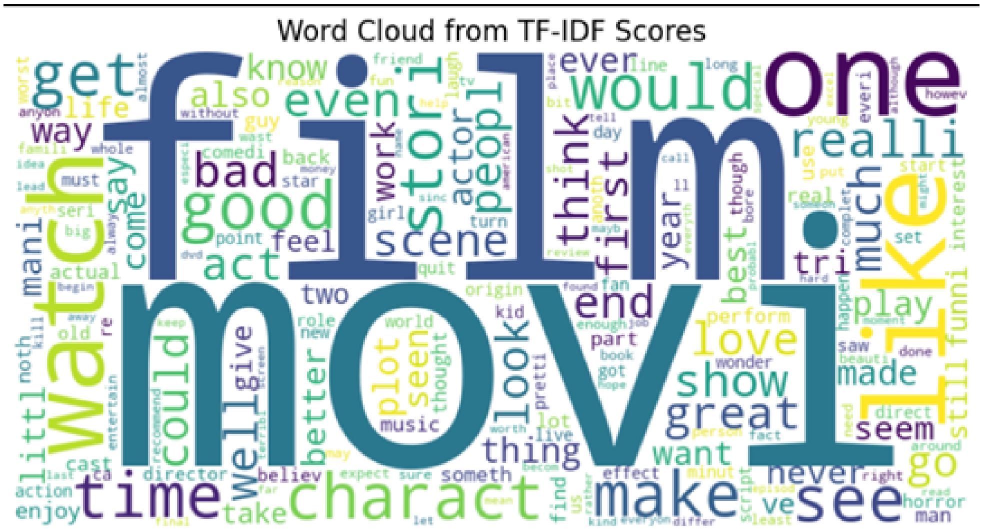
Finally, stemming is applied for each tokenized review. Stemming aims to cut off the ends of words in order to obtain correct root form. Stemming is a straightforward approach that removes common suffixes from the end of word tokens. Another option would be to use lemmatization - bringing a word to its dictionary form. For stemming we applied the **Porter Stemmer algorithm** - set of morphological rules adjusted to English (e.g. sses --> ss; ies --> i). The method is based on consonants (C) and vowels (V). A consonant is a letter other than the vowels and other than a letter “Y” preceded by a consonant. Any English word has one of the four forms given below.

- CVCV … C → collection, management
- CVCV … V → conclude, revise
- VCVC … C → entertainment, illumination
- VCVC … V → illustrate, abundance

By parameter m measure of any word is set when represented in the form **[C](VC)^m[V].** The set of rules works as follows: if m > X then suffix S1 will be replaced by stem S2 where X denotes “VC repeated X times”. The algorithm is implemented with the help of the above-mentioned NLTK library.

In order to be able to analyze text we represent it as word frequencies. For this purpose TF-IDF model is used. Term Frequency (TF) measures how often a word appears in a review, relative to the total words in data. In turn Inverse Document Frequency (IDF) measures how common a word is across all reviews. The more reviews a word appears in, the lower its IDF score, reducing its weight. Words that are rare in data get higher IDF scores. According to our assumption, rare words may have greater power to distinguish sentiment, therefore TF-IDF is used.

After calculation of word frequency, Word Cloud is built based on reviews dataset showing the frequency of words.



Top 10 words in the reviews are ‘movi’, ‘film’, ‘one’, ‘like’, watch’, ‘good’, time’, ‘see’, ‘charact’ and ‘make’.

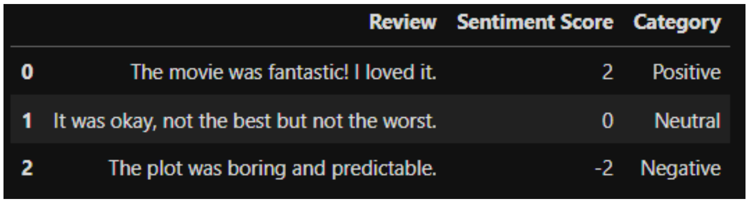
7. Dictionary-based approach

To analyze the sentiment with a dictionary we should use a predefined lexicon. Lexicon is a list of words that are associated with positive, negative, or neutral sentiments. The analysis is performed by matching words in the input review to the dictionary and aggregating the sentiment scores.

There are many already prepared lexicons and the most popular like VADE and SentiWordNet are widely used in sentiment analysis. We decided to use the NLTK opinion lexicon provided by Standford university. The choice was driven by clear structure. First, there are shortcuts to retrieve positive and negative words. Moreover, exact lists of words are given by creators on Github.

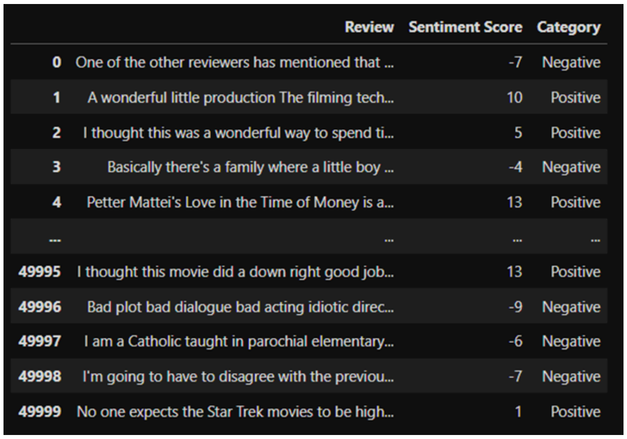
Algorithm of score assignment to a review is namely a sum calculation of categories obtained by dictionary. If positive words prevail in a single review is positive then it is labeled as “positive”. If the sum of negative words is higher then the sum of positives then the review is considered to be “negative”. If the sum is zero then there is a balance between negative and positive words and the review is “neutral”.

First we test score assignment on artificially generated reviews.

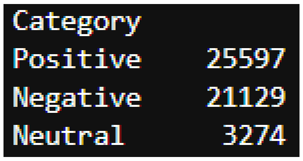


	Review	Sentiment Score	Category
0	The movie was fantastic! I loved it.	2	Positive
1	It was okay, not the best but not the worst.	0	Neutral
2	The plot was boring and predictable.	-2	Negative

After verification of the proper category assignment algorithm is implemented on the reviews dataset. It is worth mentioning that the categories count is quite consistent with the label column provided by the dataset creators.



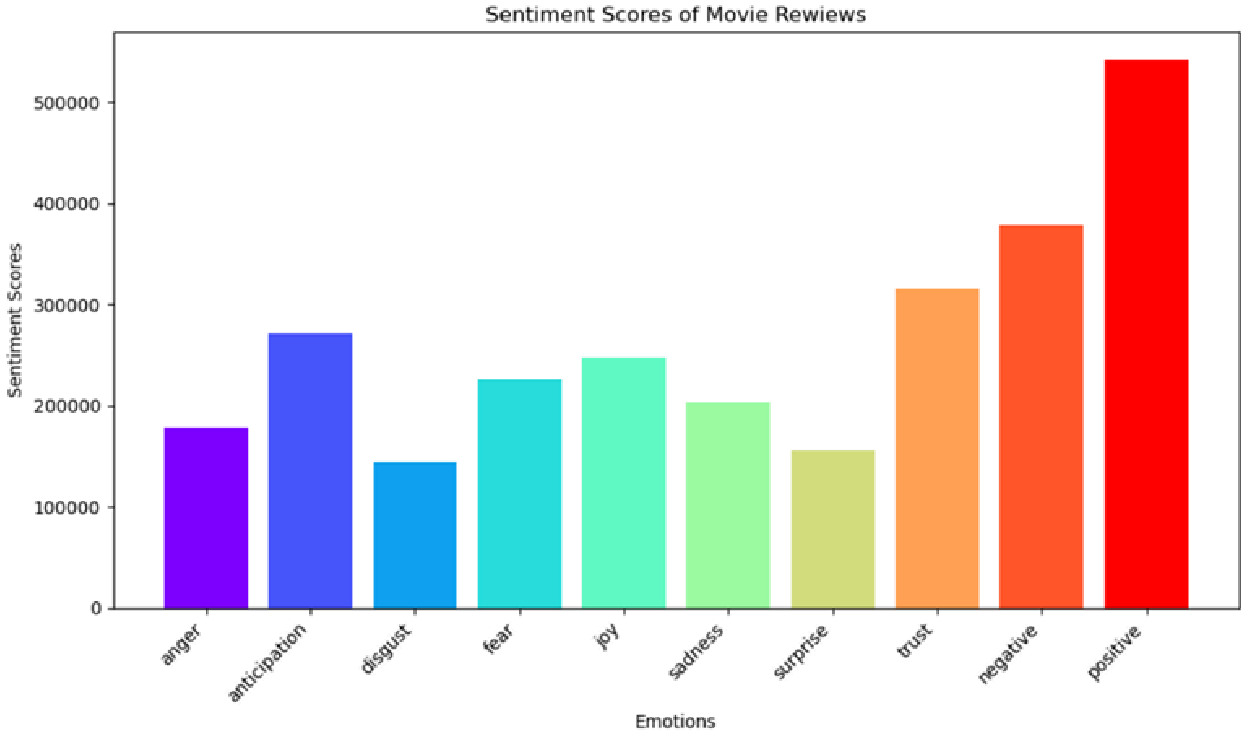
	Review	Sentiment Score	Category
0	One of the other reviewers has mentioned that ...	-7	Negative
1	A wonderful little production The filming tech...	10	Positive
2	I thought this was a wonderful way to spend ti...	5	Positive
3	Basically there's a family where a little boy ...	-4	Negative
4	Petter Mattei's Love in the Time of Money is a...	13	Positive
...	...	...	...
49995	I thought this movie did a down right good job...	13	Positive
49996	Bad plot bad dialogue bad acting idiotic direc...	-9	Negative
49997	I am a Catholic taught in parochial elementary...	-6	Negative
49998	I'm going to have to disagree with the previou...	-7	Negative
49999	No one expects the Star Trek movies to be high...	1	Positive



```
Category
Positive    25597
Negative    21129
Neutral      3274
```

Another dictionary-based approach includes an emotions range like anger, anticipation, disgust, fear, joy, sadness, surprise and trust. Prepared by National Research Council Canada emotions lexicon can help to understand not only the attitude but also coloration of a review.

The same procedure as before is used for emotion sentiment analysis. For the test reviews we confirm that there is association between words and emotions. Review data is analyzed in the same manner. Moreover, we are able to aggregate emotions statistics within the dataset.



It is observed that positive scores prevail in movie reviews. The most frequently met emotion is anticipation, while the least - disgust.

8. Modeling

In our project we decided to use traditional Machine Learning models - Logistic regression and Support Vector Machines. Neural networks like Recurrent Neural Networks (RNNs) are not used because the number of observations is not sufficient enough for neural network to be efficient in computation terms.

Application of Machine Learning models is possible since the label for a review - positive or negative - is provided by the dataset creators. The label can also be obtained based on the rating of the movie given by a voter.

8.1. Logistic regression

Since the label is a binary variable we can train logistic regression.

Application of Machine Learning models is possible since the label for a review - positive or negative - is provided by the dataset creators. The label can also be obtained based on the rating of the movie given by a voter.

According to the confusion matrix obtained from the testing part, there are 761 cases where the model incorrectly predicted review as "negative" when the actual review was "positive” (FN) and 907 cases where predicted was "positive" when in fact it is "negative" (FP).

8.2. Support Vector Classifier

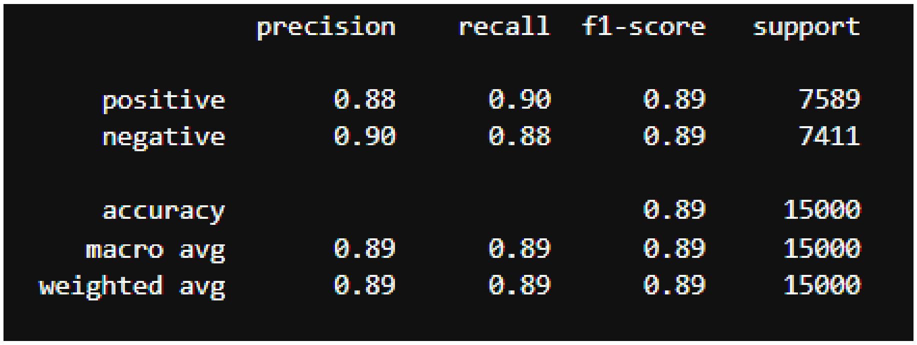
Since TF-IDF vectorization is applied to reviews, train data is high-dimensional where each feature corresponds to a unique token, i.e. word.

Support Vector Classifier is well-suited to handle high-dimensional text data. The main idea behind Support Vector Classifier is that we can use hyperplanes to create a separation between classes of positive and negative reviews. A hyperplane is chosen in a way to maximize margins between the classes allowing for misclassification by introduction of soft margin for noisy data. New prediction would be based on what side of a hyperplane the features lay on.

According to the confusion matrix obtained from the testing part, there are 686 False Negative cases and 882 False Positive cases that is better than with Logistic Regression.

8.3. Model comparison

Confusion matrices on testing data for both models are very similar as can be seen on tables below. For Support Vector Classifier precision and recall for the “positive” category is slightly better. However, given the fact that logistic regression is more interpretable and requires less computational power it is difficult to make a clear conclusion that one model is better than the other.

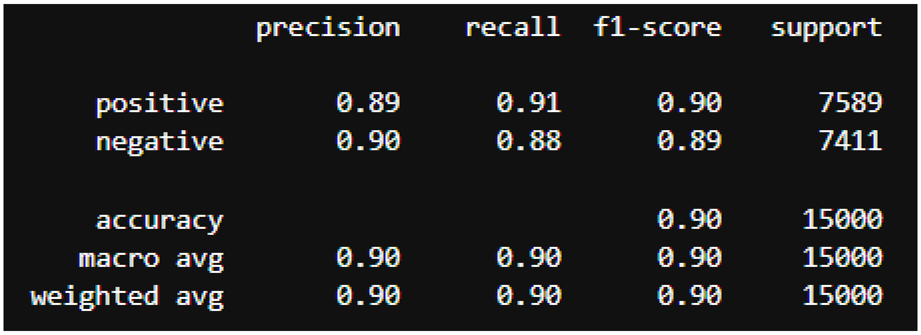


```
              precision    recall  f1-score   support

    positive       0.88      0.90      0.89      7589
    negative       0.90      0.88      0.89      7411

    accuracy                           0.89     15000
   macro avg       0.89      0.89      0.89     15000
weighted avg       0.89      0.89      0.89     15000
```

Logistic regression classification metrics



```
              precision    recall  f1-score   support

    positive       0.89      0.91      0.90      7589
    negative       0.90      0.88      0.89      7411

    accuracy                           0.90     15000
   macro avg       0.90      0.90      0.90     15000
weighted avg       0.90      0.90      0.90     15000
```

Support Vector Classifier classification metrics

For both models it was tested whether sentiment prediction is in line with expectations on unseen artificially-generated reviews. For cases where sentiment is obvious (e.g. “An absolutely fantastic film! The actors gave such powerful performances. Highly recommend it!”) both models are correct with the prediction. If sentiment is not clear even for human distinction then models behave still similarly. For instance, this mixed review is labeled as “negative” by both models.

"The movie had moments of brilliance, especially with the stunning cinematography and the lead actor's heartfelt performance. However, the plot felt unnecessarily convoluted, leaving me confused and disengaged at times. While the music score was mesmerizing and elevated some scenes, the pacing dragged in the second half, making it hard to stay invested. I appreciate the director's ambition, but the execution fell short of the emotional depth it aimed to achieve. It's neither a complete triumph nor a total disaster—just a missed opportunity."

10. Conclusions

Sentiment analysis is a useful technique for various tasks. For example, businesses can use it to analyze opinions of customers about their products to better understand their preferences and support decision making. It can be used in chatbots and voice assistants for improved understanding of human language. That way, the biggest advantage of sentiment analysis are endless possibilities to utilize it for various unusual tasks. Another one is that it avoids the subjective variability inherent in human interpretation.

However, it does have some weaknesses:

- It can struggle with sarcasm, humor, idioms, and context-specific meanings, which can lead to inaccurate sentiment classification.
- Ambiguity in sentiment can lead to incorrect interpretations. Example of such a comment may be: "The product is great, but the delivery was terrible".
- Results strongly depend on the quality of provided data (for example, if training data skews toward a certain demographic, the analysis may fail for others) and requires complex text preprocessing which may be challenging in case of some languages.

11. List of examples of the use of the method in science or practice with references to the literature

1) Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. *Learning Word Vectors for Sentiment Analysis*. The 49th Annual Meeting of the Association for Computational Linguistics. 2011.
   Source: https://ai.stanford.edu/~amaas/papers/wvSent_acl2011.pdf (access 06.11.2024)
   They are also authors of the dataset we used.
2) Muhammad Taimoor Khan, Mehr Durrani, Armughan Ali, Irum Inayat, Shehzad Khalid & Kamran Habib Khan. *Sentiment analysis and the complex natural language.* Complex Adaptive Systems Modeling. Volume 4, article number 2. 2016.
   Source: https://link.springer.com/article/10.1186/s40294-016-0016-9 (access: 10.11.2024).
3) Anuja P Jain, Padma Dandannavar. *Application of machine learning techniques to sentiment analysis.* International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT). 2016.
   Source: https://ieeexplore.ieee.org/abstract/document/7912076 (access: 15.11.2024)
4) Jacob Murel, Eda Kavlakoglu. *Stemming and lemmatization.* IBM 2023
   Source: https://www.ibm.com/topics/stemming-lemmatization (access: 15.11.2024)
5) Sentiment Analysis in R. R-bloggers. 2021.
   Source: https://www.r-bloggers.com/2021/05/sentiment-analysis-in-r-3/ (access: 15.11.2024)
6) Vijini Mallawaarachchi*. Porter Stemming Algorithm – Basic Intro*
   Source: https://vijinimallawaarachchi.com/2017/05/09/porter-stemming-algorithm/ (access: 15.11.2024)
7) Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, Jonathan Taylor. *An Introduction to Statistical Learning with Application in Python.* Springer, 2023
   Source:
   https://hastie.su.domains/ISLR2/Slides/Ch9_Support_Vector_Machines.pdf (access: 15.11.2024)